GLYFN: A GLYPH-AWARE FUSION NETWORK FOR DISTRIBUTED CHINESE EVENT DETECTION

Qi Zhai, Zhigang Kan, Linhui Feng, Linbo Qiao, and Feng Liu

College of Computer, National University of Defense Technology, ChangSha, China

ABSTRACT

Recently, Chinese event detection has attracted more and more attention. As a special kind of hieroglyphics, Chinese glyphs are semantically useful but still unexplored in this task. In this paper, we propose a novel Glyph-Aware Fusion Network, named GlyFN. It introduces the glyphs' information into the pre-trained language model representation. To obtain a better representation, we design a Vector Linear Fusion mechanism to fuse them. Specifically, it first utilizes a max-pooling to capture salient information. Then, we use the linear operation of vectors to retain unique information. Moreover, for large-scale unstructured text, we distribute the data into different clusters parallelly. Finally, we conduct extensive experiments on ACE2005 and large-scale data. Experimental results show that GlyFN obtains increases of 7.48(10.18%) and 6.17(8.7%) in the F1-score for trigger identification and classification over the state-of-the-art methods, respectively. Furthermore, the event detection task for large-scale unstructured text can be efficiently accomplished through distribution.

KEYWORDS

Distributed Chinese Event Detection, Fusion Network, Glyph.

1. INTRODUCTION

With the progress of Web2.0 interactive network and Internet technology, a variety of social software has accumulated a large amount of data. A sprawling researches have emerged about how to automatically extract event information from unstructured texts in a large number of multimodal data quickly and effectively, namely event extraction. Event detection, as a key stage of event extraction, aims at extracting event trigger words and identifying event types by triggers. For example, the sentence: "*Microsoft acquired and merged online advertisers.*" contains a "*Merge-Org*" event triggered by "*acquired*" and a "*Transfer-Ownership*" event triggered by "*merged*". In event detection task, text representation is an extremely important technical aspect, such as Xi et al. [1] find that different embeddings combinations have an important impact on the improvement of the model by replacing and combining four word vectors: character embedding, segmentation embedding, word embedding and language model embedding. Inspired by this discovery, we focus on constructing a more effective language representation in the field of event detection. It is no exaggeration to say that word embeddings have revolutionized NLP. They have experienced from the traditional high-dimensional and sparse dictionary method, one-hot and other representation methods to the dense low-dimensional distributed representation. Word2Vec [3] proposes to pre-trained the word embeddings search matrix on a large number of unlabeled corpora. It brings the combination of deep learning and natural language processing to a climax. Subsequently, many researchers begin to study it and put forward a variety of language representations [4], [5]. Recently, a novel dynamic language representation method, i.e., pre-

David C. Wyld et al. (Eds): AIAP, SIGML, CNSA, NIAI, CCSEIT - 2021
pp. 203-216, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110114

trained language model [2], [6]– [9], has been proposed, which provides better model initialization, general language representation and can effectively alleviate the problem of linguistic ambiguity. What's more, as a special kind of hieroglyphics, Chinese glyphs are semantically useful. Some researchers propose to use the image processing method to model the glyph information of the text [10].

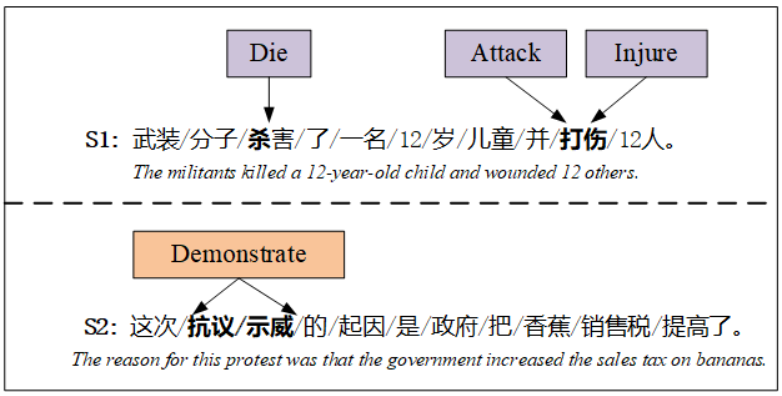


Figure 1. An example of word-triggers mismatch.

Each embedding set is trained by a different neural network. So they have different characteristics. To combine different vector information, researchers have proposed meta-embedding [11], [12] and evaluated it on text classification and other tasks. However, in the aspect of Chinese event detection, to the best of our knowledge, no researcher has integrated these different language representations. As a scene rich NLP task event detection, it is necessary to integrate the word vector diversity to learn the better-performing word. Therefore, this paper proposes a model of the Glyph-Aware Fusion Network (GlyFN) to improve the accuracy of Chinese event detection by fusing multiple text representations. Specifically, GlyFN consists of three-layer: representation layer, fusion layer, and inference layer. The representation layer contains character level glyph-aware and context-aware representations. On the one hand, the glyph-aware representation can obtain the semantic information of glyphs, on the other hand, it can effectively alleviate the mismatch problem in Chinese event detection. Taking Figure 1 as an example, in the S1 two characters in one word “打伤” (wounded) trigger two different events “Attack” and “Injure” by “打” (hit) and “伤”(wounded), respectively. While in the S2, “抗议\示威” (protest\ demonstration) is a trigger that crosses two words. Meanwhile, context-aware representation can effectively reduce ambiguity. In the fusion layer, we design an effective vector linear fusion method. Firstly, the max-pooling is used to obtain the important information of the above two representations, and then the unique features of each vector are captured by comparing the internal elements of vectors. Finally, in the inference layer, we use a multi-layer nonlinear perception to predict the values of event detection. Furthermore, to achieve the event detection of large-scale data in real events, we put forward a distributed Chinese event detection method. It consists of two components: master and worker, where the master is responsible for sending tasks and multiple workers are assigned to compute prediction results and return the results to the master. Through the mutual interaction between maser and worker, we finally implement efficient and fast event detection. In conclusion, our contributions include:

- The key contribution of this paper is the improvement to the event detection through glyph-aware fusion network. Specifically, in the representation layer, we get the glyph-aware representation through BiLSTM, and to obtain the context semantic information, we use the pre-trained language model. To fuse these two kinds of information, we propose a novel vector linear fusion mechanism. Based on acquiring the important information of common

concern among vectors, it maintains the uniqueness of each vector by comparing the internal elements of vectors, and finally obtains a better-performing fusion feature representation.

- To solve the huge computing consumption caused by massive data and large-scale models, a distributed method is proposed, which makes full use of the existing computing resources and divides the data into different nodes for parallel computing, so as to process a large number of texts and models quickly and effectively.

2. RELATED WORKS

2.1. Event Detection (ED)

ED techniques have undergone statistics-based methods, rule-based methods, traditional machine learning methods [13], and deep learning neural networks methods. Based on traditional feature representations (e.g., token, POS tags, entity information, syntactic dependency, semantic dependency, etc.) suffer from poor portability, excessive reliance on the corpus, and higher requirement-s for developers' professional fields. Besides, NLP toolkits for feature extraction may result in error propagation. With the widespread application of deep learning, based on word embeddings as feature input, event extraction via end-to-end neural networks has attracted the attention of many researchers. In the event extraction task, there exist multiple events in a sentence, and candidate arguments play different roles according to different triggers. The max-pooling of tradetional convolutional neural networks (CNN) [14] is difficult to obtain this information. Dynamic multi-pooling convolutional neural networks (DMCNN) [15] divides the text into three parts according to candidate argument and predicted trigger, and performs max-pooling on each part to solve the above problems. Due to the limited receptive field of CNN, Joint Event Extraction via Recurrent Neural Networks (JRNN) [16] uses RNN to jointly event extraction. To solve the problem of the need to design different neural networks for the feature representation of the above methods, PLMEE [17] uses the pre-trained language model for the event extraction task for the first time. EE-DGCNN [18] adopts multi-layer dilate gated CNN to reduce the number of PLMEE parameters. These methods have a good performance on English data sets. For Chinese, to obtain semantic information, Ding et al. [19] propose a Trigger-aware Lattice Neural Network (TLNN) with the addition of external language knowledge (HowNet) to enhance the understanding of ambiguous trigger words. In the feature representation, the Trigger-Aware Lattice LSTM is used to fuse the senses of characters and words to obtain the final semantic features. However, TLNN doesn't consider the full context when resolving ambiguous trigger words, and lots of external features are required for feature construction. Xi et al. [1] first propose to use BERT to obtain contextual semantic information. Specifically, it integrates word information and language model representation into the character-based model, and then inputs the final hybrid character represent-ation into BiLSTM-CRF for final ED.

2.2. Embeddings

As the most fundamental semantic unit for ED task, word embeddings are self-evident for understanding sentences and texts. Word2Vec [3] is pre-trained word embeddings search matrices on a large unlabeled corpus to obtain word vectors. Pre-trained models (PTMs) have become the basis for many practical applications in NLP and related disciplines. The development of PTMs has experienced two stages from static word embeddings to contextual word embeddings. The main representatives of static word embeddings are: NNLM [20], Word2Vec (CBOW, Skip-Gram) [3], Glove [4], FastText [5], etc. Their main flaw is that word representation is the same in different contexts. Contextual word embeddings effectively solve the above problem of ambig-uity by pre-trained encoders. Researchers have proposed the ELMo

[21] which combines the deep BiLSTM language model based on character awareness. GPT [6] is a left-to-right transformer based on the attention mechanism to generate context word vectors. But they only use one-way information. In order to fully understand the information of text context, BERT [2] dynamically obtains the contextual semantic information through a bidirectional transformer and achieves good results in 11natural language processing tasks. Meanwhile, for Chinese word embeddings, researchers have proposed a variety of methods from the Chinese characters themselves: radicals [22], strokes [23], wubi [24], glyph [10], Characters [25], etc., to obtain amore fine-grained internal semantic expression of Chinese. It can be seen from the evolution of Chinese characters that the expression of hieroglyphs is closer to the actual expression of Chinese characters. Although the simplified characters are more convenient to write, they lack the rich glyph of Chinese characters, so the semantic information of Chinese characters can be obtained more fully by encoding the pictographs of Chinese characters.

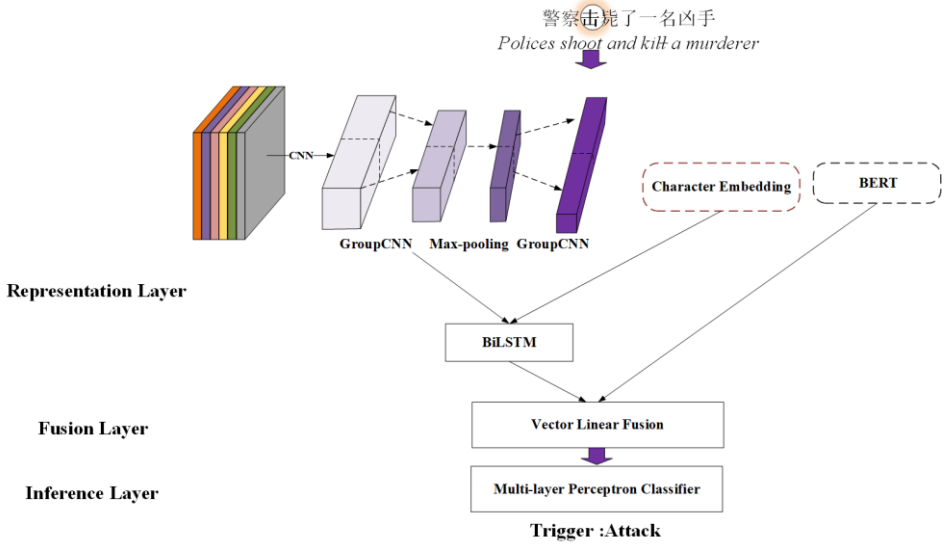


Figure 2. The architecture of GlyFN.

2.3. Meta-embeddings

Recently, several researchers have combined multiple word embeddings in the text representation phase. CharWNN [26] combines character-based and word-based embeddings. Glyce [10] is to concatenated glyph representation with BERT. Yin and Sch¨utze [11]first propose fusing word embeddings via neural networks. It merges multiple word embeddings into one word embedding through four methods: CONC, SVD, 1TON, and 1TON$^+$. However, in the above method, the fusion of word embeddings is regarded as a pre-processing step. It doesn't dynamically adapt to specific tasks. DME [12] dynamically obtains the weight value of each word embedding based on sentence-level self-attention, and acquires the final sentence representation through the BiLSTM-Max encoder. When the word embeddings of DME are input to the attention mechanism, each element inside the word embeddings is independently linear regression. It does not take into account the interaction between the elements inside the word embedding. Based on DME, DTFME [27] increases the internal relations of word embeddings through factorization and pooling operations. Auto encoder has been widely used as a learning method of feature representation. It can reconstruct the input from potentially noisy inputs through nonlinear changes. The middle one can capture important input information. AEME [28] construct the corresponding encoder and decode according to the different input word embeddings and then obtained a common and consistent word embedding space representation, that is, meta-embeddings.

3. METHODOLOGY

Given a sentence, GlyFN is to select the most appropriate assignment T to each character C in t-he untagged text S from a predefined set of tags or labels U, i.e., for the event detection system, its input is $S=\{c_1,c_2,...,c_n\}$, and the output is a prediction $T \in U$. More formally, the task is to learn a sequence labeling function:

$$F(S) \rightarrow T.$$

3.1. Technical Details of GlyFN Model

The overall framework of our model is shown in Figure 2. The whole model consists of three parts: (1) **Representation Layer**: aims to obtain the context representation of input text. According to the different granularity of input, the representation layer is divided into two representations, one is based on the glyph, the other is based on character. (2) **Fusion Layer**: mainly tac-kles the problem of integrating the two heterogeneous context representations of the Represent-ation Layer into a more complete feature representation of semantic information, so as to extract events more accurately. (3) **Inference Layer:** calculates the probability of being a trigger for each character in the input text. We then examine each layer in detail and give intuitions about its formulation.

3.1.1. Representation Layer

For the text S, it contains n characters, where c_i represents the ith character in the sequence. In the glyph level, each character c_i is first initialized to a grayscale. To represent the grayscale in a continuous space, we encode the grayscale with Group CNN [10] as shown in Figure 2. It groups the channels and convolutes them separately to reduce the parameters. Then we obtain the glyph feature representation of each character $G_i \in R^{d_1}$. In addition, we utilize the pre-trained represent-ation of each character $P_i \in R^{d_2}$ to further enrich the lexical information of the glyph level. Let X_i be the concatenation of the two representations:

$$X_i = G_i \oplus P_i, \tag{1}$$

Where $X_i \in R^{d_1+d_2}$. To further obtain a glyph-aware contextual representation, the bidirectional LSTM (i.e., BiLSTM) with two layers is introduced to capture contextual semantic information. Formally, given the feature embeddings sequence of the sentence $S'=\{X_1,X_2,...,X_n\}$, BiLSTM computes two sets of n hidden states, one for each direction:

$$\begin{aligned} \overrightarrow{h_i} &= LSTM\left(\overrightarrow{h_{i-1}}, X_i\right), \\ \overleftarrow{h_i} &= LSTM\left(\overleftarrow{h_{i+1}}, X_i\right), \\ H_i &= \left[\overrightarrow{h_i}; \overleftarrow{h_i}\right], \end{aligned} \tag{2}$$

where $H_i \in R^d$ is the final hidden state of the BiLSTM at the ith step, which is the concatenation of $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$. At the character level, to gain context-aware representation, we incorporate the pre-trained language model embedding from BERT [2] into character representation. The structure of BERT can not only deal with the problem of long-distance dependence but also capture the

bidirectional feature of context, which can effectively alleviate the ambiguity problem. In view of these advantages, a pre-trained BERT is introduced to generate context-aware representation:

$$B = E(S). \tag{3}$$

Here E refers to the BERT encoder model, $B \in R^{n \times d}$ is the last layer hidden output, and $B_i \in R^d$ represents the ith output of the sequence S.

3.1.2. Fusion Layer

Its purpose is to generate a complete representation by integrating glyph awareness and context information. As mentioned above, we set the vector dimension of the two representations to the same dimension d. Thus they are in the same vector space. Specifically, we design four different paradigms to fuse them up: Simple Combination, Linear Attention, Gate Mechanism, Vector Linear Fusion.

Simple Combination contains concatenation, summation, aver-pooling, and max-pooling four sensible strategies for combining two representations:

$$\begin{aligned} F_i^{Cat} &= B_i \oplus H_i, \\ F_i^{Sum} &= B_i + H_i, \\ F_i^{Avg} &= averpooling([B_i, H_i]), \\ F_i^{Max} &= maxpooling([B_i, H_i]), \end{aligned} \tag{4}$$

Where $F_i^{Cat} \in R^{2d}$, $F_i^{Sum} \in R^d$, $F_i^{Avg} \in R^d$, and $F_i^{Max} \in R^d$don't introduce any additional parameter. What's more, $F_i^{Sum} \in R^d$ is the most effective in our experiments.

Linear Attention aims to dynamically generate different weights of embeddings in the process of training. We combine the two representations by taking the weighted sum:

$$F_i^{Att} = \sum_{k=1}^{2} \lambda_{k,i} V_{k,i}, \tag{5}$$

Where $V_{k,i}$ consists of B_i and H_i, $\lambda_{k,i}=f(\{V_{k,i}\}_{i=1}^{n})$ are scalar weights from the linear attention:

$$\lambda_{k,i} = f(V_{k,i}) = \sigma(\max(0, V_{k,i}W^1 + b^1)W^2 + b^2), \tag{6}$$

where σ is a *softmax (·)* function, $W^1 \in R^{d \times \frac{d}{2}}$, $b^1 \in R^{\frac{d}{2}}$, $W^2 \in R^{\frac{d}{2} \times 1}$, $b^2 \in R$ are the weights and bias, respectively .$\lambda_{k,i}$ represents the contribution of B_i and H_i, to the final fusion representation F_i^{Att}, which models the importance of individual features in the given contexts.

Gate Mechanism Different parts of the text sequence have different meanings and importance. To obtain the weight of each character dynamically, we assign different weight values to the two embeddings corresponding to each character:

$$
\begin{aligned}
H' &= \delta(H), \\
B' &= \delta(B), \\
H^{gate} &= H^{1'} \times softmax(H^{2'}), \\
B^{gate} &= B^{1'} \times softmax(B^{2'}), \\
W^{H} &= maxpooling(H^{gate} \times B^{gate}), \\
W^{B} &= maxpooling(B^{gate} \times H^{gate}), \\
\alpha_H, \alpha_B &= f(\{c_i\}_{i=1}^{n}) = softmax(W^{H}, W^{B}), \\
F^{Gate} &= \alpha_H \times H + \alpha_B \times B,
\end{aligned}
\tag{7}
$$

where δ is a $GELU(\cdot)$ function, $H^{1'}$, $H^{2'} \in R^{n \times \frac{d}{4}}$ are half of H', $B^{1'}, B^{2'} \in R^{n \times \frac{d}{4}}$ are also half of B'. To verify and test the effectiveness of the distribution, we used the more complex gate mechanism F^{Gate} of the fusion network in our distributed experiments.

Vector Linear Fusion As the Linear Attention uses neural network for fusion, it cannot clearly represent the common focus of the fusion vector and retain its own unique information. Thus, to tackle this problem, we propose the Vector Linear Fusion. What's more, it considers the role of elements inside a vector when fusing. Mathematically, we first apply a *max* operation over the two representations to get the common important information:

$$
F'_i = max([B_i, H_i]). \tag{8}
$$

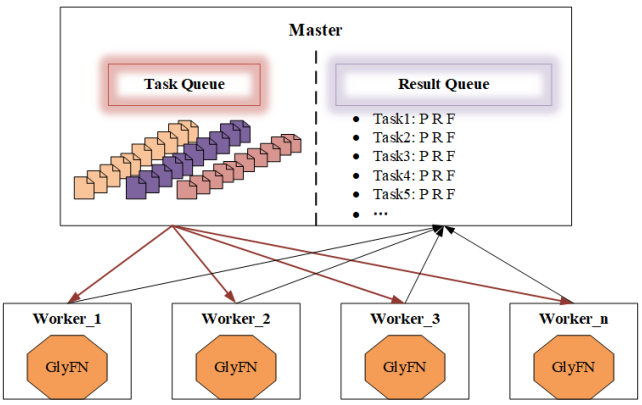


Figure 3. Distributed Chinese Event Detection.

To retain the unique information of the two representations, and integrate the information of the inner elements of each vector, we compared them with F'_i, respectively:

$$
\begin{aligned}
B'_i &= B_i - F_i, \\
H'_i &= H_i - F_i.
\end{aligned}
\tag{9}
$$

Finally, we integrate these three representations whereby the addition operation is performed:

$$
F_i^{fusion} = F'_i + B'_i + H'_i, \tag{10}
$$

where F_i^{fusion} is the ultimate fusion expression, the way of agreeing to disagree, which is the most effective method of the experiments.

3.1.3. Inference Layer

The goal of inference layer is to calculate the event category probability of each character for the input event mention. To achieve this, we take the final fusion representation in Eq. (10) as the input into the two-layer fully-connected neural network and *softmax (·)* function:

$$P = argmax\left(\sigma\left(\max\left(0, F^{fusion} W_1 + b_1\right) W_2 + b_2\right)\right), \tag{11}$$

where $P \in R^{n \times 1}$ is the output vector, σ is the non-linear activation *softmax (·)* function, and $W_1 \in R^{d \times \frac{d}{2}}, W_2 \in R^{\frac{d}{2} \times U}$ (U is the number of event types) are the weight matrix for dimension transformation. $b_1 \in R^{\frac{d}{2}}$ and $b_2 \in R^U$ are the biases. We train our model through the cross-entropy error function, and the purpose is to minimize the following *Loss*:

$$Loss(P, Y) = -\sum_{i=1}^{n} \sum_{k=1}^{U} y_{ik} \cdot \log(p_{ik}), \tag{12}$$

Where n is the number of characters in the input sequence S, U is the total number of event types, and y_{ik} is 1 if the character $c_i$belongs to the class k. To optimize the parameters, we use *Adam* as optimizer to update the parameters of GlyFN.

3.2.Distributed Chinese Event Detection

With the extensive application of deep learning in many fields of natural language processing, the scale of models is constantly increasing, and the amount of data of models is also exploding. Asingle CPU card, or multiple GPU cards on a single server, is no longer sufficient for training tas-ks. Therefore, the efficiency of distributed training, that is, the use of multiple servers for collaborative training, has become a necessary choice for large-scale data training. In the paper, we trained the detection of events in a distributed way. As shown in Figure 3, we divide the computers in the cluster into two categories.

Table 1. Results of our GlyFN against baseline methods (\%).

Model	**Trigger Identification**			**Trigger Classification**		
	P	R	F1	P	R	F1
Char-based C-BiLSTM [30]	65.6	66.7	66.1	60	60.9	60.4
Word-based C-BiLSTM [30]	75.8	59	66.4	69.8	54.2	61
HNN [31]	74.2	63.1	68.2	**77.1**	53.1	63
NPN [29]	64.8	73.8	69	60.9	69.3	64.8
TLNN [19]	67.34	74.68	70.82	64.45	71.47	67.78
BiLSTM+CRF(char+Im+seg+word) [1]	68.9	78.8	73.5	66.4	76	70.9
GlyFN (ours)	**80.32**	**81.64**	**80.98**	76.45	**77.70**	**77.07**

Table 2. Effect of Fusion Methods (\%)..

Model	**Trigger Identification**			**Trigger Classification**		
	P	R	F1	P	R	F1
BERT	74.27	83.28	78.52	70.47	79.02	74.50
Glyph	66.03	68.20	67.10	63.49	65.57	64.52
Concatenate	75.59	84.26	79.69	71.76	**80.00**	75.66
Avrg-pooling	78.70	83.61	**81.08**	73.77	78.36	75.99
Sum	76.97	83.28	80.00	73.03	79.02	75.91
Max-pooling	77.71	**84.59**	81.00	72.89	79.34	75.98
Liner-attention	71.22	76.26	73.65	67.06	71.80	69.34
Gate Mechanism	78.30	81.64	79.94	74.53	77.70	76.08
GlyFN (ours)	**80.32**	81.64	80.98	**76.45**	77.70	**77.07**

3.2.1. Master

In order to communicate with multiple processes, the master creates a task queue and result queue. Aimed to ensure the communication between the master and multiple processes, their server address and port must be consistent. The master is responsible for assigning tasks to multiple processes and getting results from the result queue via network communication. After all tasks are completed and results are obtained, the worker process is closed and then the master is closed.

3.2.2. Worker

The goal of workers is to deal with the tasks assigned by the master. The workers also set up the task queue and the result queue. Specifically, the worker retrieves the task to be processed from the task queue, then detects the event through the GlyFN model. Finally, it returns predicted results of input sequences to the result queue.

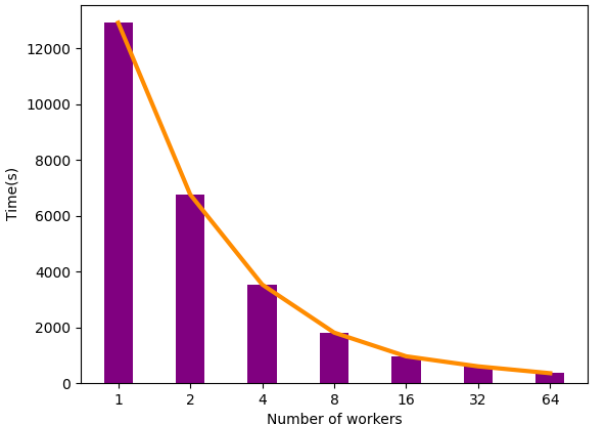


Figure 4. The time consumption of the ED task in different computing workers.

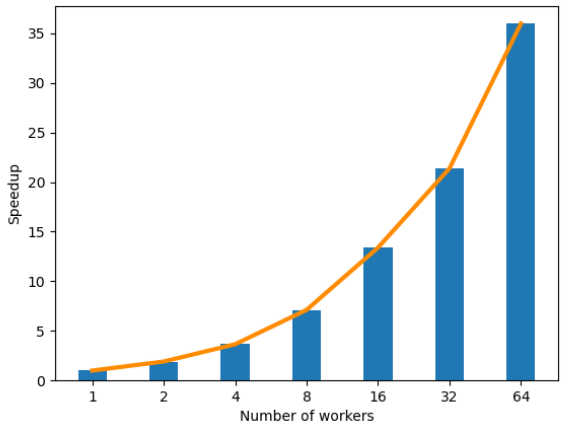


Figure 5. The speedup of the ED task in different computing workers.

4. EXPERIMENT

We evaluate the proposed GlyFN model using the ACE2005 Chinese corpus and compare it with baselines. Our goal is to verify whether the proposed model is effective for the event detection. The dataset contains 633 Chinese documents with 33 gold standard labels. We use the same setup as Chen and Ji [13], lin et al. [29] and Xi et al. [1]. Specifically, 569/64/64 documents are used as training/validation/test set, respectively. On evaluation metrics, we use the standard micro-averaged precision (P), recall (R), and F1-score (F1) for event detection. Besides only when the trigger words type and offset are exactly the same as the label, the trigger word can be recognized correctly [13].

4.1. Implementation Details

In our experiments, the embedding sizes of characters, glyph and BERT are 50/128/768, respectively. To be the same dimension as the BERT embedding, we set the dimension of the hidden vector of the BiLSTM to 384. In addition, we set the learning rate to 0.00001 for context-aware representation and 0.001 for glyph-aware representation. Furthermore, we option the batch size to 64 and the epoch of training process to 200. Finally, we artificially add a “NULL” label to map non-trigger items and a “PAD” label to map padding items. In conclusion, the total number of event types is 35.

4.2. Overall Results

The experimental results are shown in Table 1. We can make the following observations:

Table 3. Efficiency of different workers

Workers	1	2	4	8	16	32	64
Efficiency	1	0.95	0.91	0.89	0.84	0.67	0.56

(1) By comparing the experimental results of baselines, we can find that GlyFN achieves the best performance on ACE 2005 Chinese corpus. It achieves 7.48 (10.18\%) and 6.17 (8.7\%) F1-score improvements on trigger identification and classification respectively. This demonstrates that glyph-aware representation can improve the accuracy of identifying trigger words by obtaining sequence order information. Further, thanks to the fusion network, the fusion representation

greatly improves the accuracy of event detection. (2)Chinese glyph information plays an important role in semantic representation in terms of Chinese event detection. Compared with the model *BiLSTM+CRF(char+Im+seg+word)* , which takes character embedding, BERT embedding, segmentation embedding, and word embedding as input, GlyFN which contains glyph information achieves the better performance.

4.3. Effect of Fusion Methods

This part is to analyze the effectiveness of fusion methods. Different word embeddings adopt different encoding methods, and the information obtained is complementary. Therefore, we propose to get richer semantic information through fusion. We performed the following ablation experiments. In Table 2, we observe that almost all of the fusion methods have a higher F1-score than the embedding alone, indicating that we can achieve a more adaptive event detection representation by fusion. At the same time, we find that the integration of *Liner-attention* doesn't perform well. Our explanation for the inferior performance of the *Liner-attention* strategy is as follows: through the attention mechanism, it gives weight to the words in the sentence directly, without considering the information of the internal elements of the embedding, and fails to highlight the characteristics of each representation. Compared with *Max-pooling*, our method GlyFN is obviously more advantageous in the classification of trigger words, which indicates that the unique characteristics of each embedding should be retained during the integration.

4.4. Evaluation of the Distributed System

As shown in Figure 3, we implement distributed operation of event detection through interaction between master and worker. In order to better evaluate the performance of distributed computing, we analyze the running time, speedup, and efficiency under the different numbers of workers. From Figure 4, we can observe that the running time of the model decreases with the increase of workers. Specifically, as the number of workers increased, the corresponding elapsed time decreased by nearly half, with one worker taking almost 36 times as long as 64 workers. We also compare the speedup ratios of different workers. The speedup ratio is a measure of the performance and effect of the parallel system or program parallelization. As shown in the Figure 5, as the number of workers increases, the speedup ratio decreases exponentially, which demonstrates the effectiveness of the parallel algorithm for a large number of event detection data. In addition, the efficiency can represent the utilization rate of each processor in the system. Table 3 shows the efficiency of different workers. With the increase of workers, the efficiency of each processor is decreasing, which means that it is possible to process massive data. Therefore, the distributed application to the event detection task can greatly shorten the running time, and can quickly and effectively deal with large-scale model and data.

5. CONCLUSION

In this paper, we propose a novel Glyph-Aware Fusion Network (GlyFN) to improve the Chinese event detection with a special focus on the utilization of glyphs. During the experiments, we found that glyph information plays an important role in semantic representation. In addition, GlyFN uses the vector linear fusion method to combine context-aware representation and glyph-aware representation, which can not only obtain common important information among different representations but also maintain uniqueness by comparing the internal elements of the vectors. Through a large number of comparative experiments, we showed the effectiveness of the fusion method. Further, for large-scale data in the real world, we propose a distributed event detection method. Through master-worker interaction, efficient and fast event detection can be achieved. In

the future, we will explore more Chinese information, such as word, and combine with different pre-trained language models to further improve the accuracy of event detection.

ACKNOWLEDGEMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0204301, and in part by the National Natural Science Foundation of China under Grant 61806216.

REFERENCES

[1] X. Xiangyu, Z. Tong, Y. Wei, Z. Jinglei, X. Rui, and Z. Shikun, "A hybrid character representation for chinese event detection," in 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp. 1–8.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprintarXiv:1810.04805, 2018.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[4] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

[6] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[7] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, "Ernie: Enhanced representation through knowledge integration," arXiv preprint arXiv:1904.09223, 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv: 1907.11692, 2019.

[9] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in Advances in neural information processing systems, 2019, pp. 5753–5763.

[10] Y. Meng, W. Wu, F. Wang, X. Li, P. Nie, F. Yin, M. Li, Q. Han, X. Sun, and J. Li, "Glyce: Glyph-vectors for chinese character representations," in Advances in Neural Information Processi-ng Systems, 2019, pp. 2746–2757.

[11] C. N. d. Santos and V. Guimaraes, "Boosting named entity recognition with neural character embeddings," arXiv preprint arXiv:1505.05008, 2015.

[12] D. Kiela, C. Wang, and K. Cho, "Dynamic meta-embeddings for improved sentence representatio-ns," arXiv preprint arXiv:1804.07983, 2018.

[13] Z. Chen and H. Ji, "Language specific issue and feature exploration in chinese event extraction," in Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, 2009, pp.209–212.

[14] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers),2015, pp. 365–371.

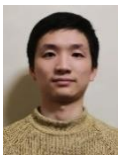
[15] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in Proceedings of the 53rdAnnual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume1: Long Papers), 2015, pp. 167–176.

[16] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 300–309.

[17] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, "Exploring pre-trained language models for event extraction and generation," in Proceedings of the 57thAnnual Meeting of the Association for Computational Linguistics, 2019, pp. 5284–5294.
[18] Z. Kan, L. Qiao, S. Yang, F. Liu, and F. Huang, "Event arguments extraction via dilate gated convolutional neural network with enhanced local features," arXiv preprint arXiv:2006.01854, 2020.
[19] N. Ding, Z. Li, Z. Liu, H. Zheng, and Z. Lin, "Event detection with trigger-aware lattice neural network," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),2019, pp. 347–356.
[20] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," Journal of machine learning research, vol. 3, no. Feb, pp.1137–1155, 2003.
[21] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," arXiv preprintarXiv:1802.05365, 2018.
[22] Y. Li, W. Li, F. Sun, and S. Li, "Component-enhanced chinese character embeddings," arXiv preprint arXiv:1508.06669, 2015.
[23] S. Cao, W. Lu, J. Zhou, and X. Li, "cw2vec: Learning chinese word embeddings with stroke n-gram information." In AAAI, 2018, pp. 5053–5061.
[24] M. X. Tan, Y. Hu, N. I. Nikolov, and R. H. Hahnloser, "wubi2en: Character-level chinese-english translation through ascii encoding," arXiv preprintarXiv:1805.03330, 2018.
[25] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.
26] C. Dos Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in International Conference on Machine Learning. PMLR, 2014, pp. 1818–1826.
[27] Y. Xie, Y. Hu, L. Xing, and X. Wei, "Dynamic task-specific factors for meta-embedding," in International Conference on Knowledge Science, Engineering and Management. Springer, 2019, pp. 63–74.
[28] D. Bollegala and C. Bao, "Learning word meta-embeddings by autoencoding," in Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1650–1661.
[29] H. Lin, Y. Lu, X. Han, and L. Sun, "Nugget proposal networks for chinese event detection," arXiv preprint arXiv:1805.00249, 2018.
[30] Y. Zeng, H. Yang, Y. Feng, Z. Wang, and D. Zhao, "A convolution bilstm neural network model for chinese event extraction," in Natural Language Understanding and Intelligent Applications. Springer, 2016, pp. 275–287.
[31] X. Feng, B. Qin, and T. Liu, "A language-independent neural network for event detection," Science China Information Sciences, vol. 61, no. 9, p. 092106,2018.

AUTHORS

Qi Zhai received her B.E. degree in computer science from the National University of Defense Technology (NUDT), Changsha, China, in 2018. Her research interests include event extraction and text representation.



Zhigang Kan received his B.E. and M.E. degree in computer science from the National University of Defense Technology (NUDT), Changsha, China, in 2017 and 2019, respectively. He is currently doing a doctorate in computer science. His research interests include event extraction and even tic graph.

Linhui Feng received her B.E. degree in computer science from the National University of Defense Technology (NUDT), Changsha, China, in 2019. Her research interests include event extraction and few-shot learning.

Linbo Qiao received the Ph.D., M.S., and B.S. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, China, in 2017, 2012 and 2010, respectively. Now, he is an assistant research fellow in the National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, P.R. China. He worked as a research assistant in Chinese University of Hong Kong, from May 2014 to Oct. 2014. His research interests include structured sparse learning, online and distributed optimization, and deep learning for graph and graphical models.

Feng Liu received the Ph.D., M.S., and B.S. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, China, in 2006, 2002 and 1999, respectively. Now, he is an associate research fellow in the National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, P.R. China. His research interests include distributed computing, big data.